

**AMENDMENTS TO THE CLAIMS**

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (original) A method of dividing a task amongst a plurality of nodes within a distributed computer, said method comprising:

receiving requirements data indicating desired properties of a task group of nodes and interconnections between them, which properties lead to said task group being suited to said task or tasks of a similar type;

calculating a task group topology in dependence upon said requirements data;  
and

distributing said task amongst the plurality of nodes in accordance with the task group topology thus calculated.

2. (original) A method according to claim 1 wherein said topology calculation comprises the step of comparing said requirements data with node capability data for a node available to join said task group.

3. (original) A method according to claim 2 wherein said requirements data comprises one or more property value pairs.

4. (original) A method according to claim 3 wherein said requirements data is arranged in accordance with a predefined data structure defined by requirements format

data stored in said computer, said method further comprising the step of verifying that said requirements data is formatted in accordance with predefined data structure by comparing said requirements data to said requirements format data.

5. (currently amended) A method according to ~~any preceding claim~~ claim 1 wherein said node capability data comprises one or more property value pairs.

6. (original) A method according to claim 5 wherein said node capability data is arranged in accordance with a predefined data structure defined by node capability format data stored in said computer, said method further comprising the step of verifying that said node capability data is formatted in accordance with predefined data structure by comparing said node capability data to said node capability format data.

7. (currently amended) A method according to ~~any preceding claim~~ claim 1 further comprising the step of operating a node seeking to join said task group to generate node capability data and send said data to one or more nodes already included within said task group.

8. (currently amended) A method according to ~~any preceding claim~~ claim 1 wherein said task distribution involves a node forwarding a task to a node which neighbours it in said task group topology.

9. (original) A method according to claim 1 wherein said requirements data comprises data relating to the amount of data storage or processing power available at said node.

10. (original) A method according to claim 1 wherein said requirements data comprises data relating to the quality of communication between said node and one or more nodes already selected for said task group.

11. (original) Distributed computer apparatus comprising: a plurality of data processor nodes, each connected to at least one other of said data processor nodes via a communications link; each of said nodes having recorded therein:

- a) group membership policy data;
- b) a list of group members;
- c) processor readable code executable to update group membership data, said code comprising:

- group membership request generation code executable to generate and send a group membership request including node profile data to another node indicated to be a member of said group;

- group membership request handling code executable to receive a group membership request including node profile data, and decide whether said request is to be granted in dependence upon the group membership policy data stored at said node;

**MCKEE**  
**U.S. National Phase of PCT/GB2003/002631**

group membership update code executable to update the list of group members stored at said node on deciding to grant a group membership request received from another node, and to send a response to the node sending said request indicating that said request is successful.

12. (original) Distributed computer apparatus according to claim 11, wherein each node further has recorded therein node profile data generation code executable to generate said node profile data.

13. (currently amended) Distibruited Distributed computer apparatus according to claim 11 ~~or claim 12~~, wherein each node further has recorded therein group membership policy data distribution code executable to distribute said policy data, said policy distribution code comprising: policy input code operable to receive policy data; policy storage code operable to store said received policy data at said node; and policy forwarding code operable forward said policy from said node to at least one other node in said distributed computer apparatus.

14. (currently amended) Distributed computer apparatus according to ~~any one of claims 11 to 13~~ claim 11, wherein each node further has recorded therein policy format data; and policy data format verification code executable to check that said received policy data accords with said policy format data.

**MCKEE**  
**U.S. National Phase of PCT/GB2003/002631**

15. (currently amended) Distributed computer apparatus according to ~~any one of claims 11 to 14~~ claim 11, wherein each node further has recorded therein profile format data; and profile data format verification code executable to check that said received node profile data accords with said profile format data.

16. (currently amended) Distributed computer apparatus according to ~~any one of claims 11 to 14~~ claim 11, wherein each node further has recorded therein received program data execution code executable to receive program data from another of said nodes and to execute said program.

17. (original) Distributed computer apparatus according to claim 16, wherein said plurality of processor nodes comprise computers executing different operating systems programs, and said received program execution code is further executable to provide a similar execution environment on nodes despite the differences in said operating system programs.

18. (original) A method of operating a member node of a distributed computing network, said method comprising: accessing membership policy data comprising one or more property value pairs indicating one or more criteria for membership of said distributed computing network; receiving, from an applicant node, profile data comprising one or more property value pairs indicating characteristics of the applicant node; determining whether said applicant profile data indicates that said applicant node

meets said membership criteria; responsive to said determination indicating that said applicant node meets said membership criteria, updating distributed computing network membership data accessible to said member node to indicate that said applicant node is a member node of said distributed computing network.

19. (original) A method according to claim 18 wherein said member node stores said distributed computing network membership data.

20. (original) A method according to claim 19 wherein said member node stores said membership policy data.

21. (original) A method according to claim 20 further comprising the steps of: updating said membership policy data; removing indications that one or more nodes are members of said distributed computing network from said distributed computing network membership data; sending an indication to said one or more nodes requesting them to re-send said profile data.

22. (original) A computer program product loadable into the internal memory of a digital computer comprising: task group requirements data reception code executable to receive and store received task group requirements data; node capability profile data reception code executable to receive and store received node capability profile data; comparison code executable to compare said node capability data and said task group

requirements data to find whether the node represented by said node capability data meets said task group requirements; task group topology update code executable to add an identifier of said represented node to a task group topology data structure on said comparison code indicating that said represented node meets said requirements; task execution code executable to receive code from another node in said task group and to execute said code or forward said code to a node represented as a neighbour in said task group topology data structure.

23. (original) A method of operating a network to create a logical network topology based on the physical topology of said network, said logical network topology being suited to a task, said method comprising: identifying a member node as a member of said logical network; storing requirements data representing what is required of nodes in order for them to be suitable for said task; storing candidate node capability data representing the capabilities of a candidate node in said physical network; operating a candidate node in said network to compare its candidate node capability data with said requirements data; responsive to said comparison indicating that said candidate node to meet said requirements, making said node a member of said logical network.